Systèmes linéaires

$$\mathrm{A}\,\boldsymbol{x} = \boldsymbol{b}$$

Conditionnement d'un système linéaire

- Si A est une matrice inversible, le conditionnement relatif du système est

$$K_{\mathrm{rel}}(\boldsymbol{b}) \leq \mathrm{K}(\mathrm{A}) = \left|\mathrm{A}^{-1}\right|\left|\mathrm{A}\right|$$

K (A) = conditionnement de la matrice

- Rappel : norme d'une matrice

$$\left|\mathrm{A}\right| = \sup_{|\boldsymbol{x}| \neq 0} \frac{\left|\mathrm{A}\,\boldsymbol{x}\right|}{\left|\boldsymbol{x}\right|} = \sup_{|\boldsymbol{x}| = 1} \left|\mathrm{A}\,\boldsymbol{x}\right|$$

Comment résoudre le système ?

- Solution 1
  - Calculer la matrice inverse
  - Faire la multiplication
  - Coût de calcul ?
- Solution 2
  - Pivot de Gauss
  - Coût de calcul ?
- Matrices particulières
  - Triangulaires
  - Diagonales

Différents types de matrices

- Pleines
  - Très peu de valeurs nulles
  - En général, il en résulte un grand coût de calcul

- Creuses (sparse)
  - Beaucoup de valeurs nulles
  - Ex : diagonales
  - Systèmes plus rapides a résoudre

A est une matrice diagonale

problème

$$\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \ddots & \vdots \\ 0 & 0 & a_{ii} & 0 & 0 \\ \vdots & \ddots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 0 & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{pmatrix}$$

solution

$$x_i = \frac{b_i}{a_{ii}} \quad , \quad i \in [1, n]$$

Fonction $x = \text{diago}(A,b)$

Algorithme

pour $i = 1$ jusqu'à n

$$x_i \leftarrow \frac{b_i}{a_{ii}}$$

fait

A est de forme triangulaire

$$\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \ddots & \vdots \\ a_{i1} & \ddots & a_{ii} & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ a_{n1} & \cdots & a_{ni} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{pmatrix}$$

$$\begin{cases} x_1 = \dfrac{b_1}{a_1} \\ x_i = \dfrac{1}{a_{ii}} \underbrace{\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right)}_{\text{somme}} \end{cases}$$

Fonction $x = \text{triang}(A,b)$

Algorithme

$$x_1 \leftarrow \frac{b_1}{a_{11}}$$

pour $i = 2$ jusqu'à n

  somme $\leftarrow b_i$

  pour $j = 1$ jusqu'à $i - 1$

    somme $\leftarrow$ somme $- a_{ij} x_j$

  fait

  $x_i \leftarrow \dfrac{\text{somme}}{a_{ii}}$

fait

Deux familles de méthodes

- Méthodes directes
  - Triangulation et/ou factorisation
  - Pivot de Gauss
  - Factorisation LU
  - Factorisation de Cholesky
  - Factorisation de Householder et QR

- Méthodes itératives

Factorisation LU

- A est inversible
- Trouver L et U, matrices triangulaires inférieure et supérieure tel que

$$\mathrm{A} = \mathrm{L}\,\mathrm{U}$$

- Contexte d'utilisation
  - $\mathbf{b}$ change mais pas A

$$\mathrm{L}\,\boldsymbol{y} = \boldsymbol{b} \text{ puis } \mathrm{U}\,\boldsymbol{x} = \boldsymbol{y}$$

Factorisation LU

- Décomposition, unique, existe ssi
  - les n mineurs de A sont non nuls.
- U : méthode d'élimination de Gauss
- L : pivot successifs de Gauss
- Coût : $2/3\ n^3$
- Instable sans permutation

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_2^1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n-1}^1 & l_{n-1}^2 & \cdots & 1 & 0 \\ l_n^1 & l_n^2 & \cdots & l_n^{n-1} & 1 \end{bmatrix}$$

Factorisation de Cholesky

- A est définie positive et symétrique $\quad \forall \boldsymbol{x} \quad \langle \mathrm{A}\,\boldsymbol{x} \cdot \boldsymbol{x} \rangle \geq 0$
- Il existe une unique matrice R triangulaire inférieure telle que

$$\mathrm{A} = \mathrm{R}\,\mathrm{R}^{\mathrm{T}}$$

- Contexte d'utilisation
  - $\mathbf{b}$ change mais pas A

$$\mathrm{R}\,\boldsymbol{y} = \boldsymbol{b} \text{ puis } \mathrm{R}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{y}$$

Factorisation de Cholesky

- Cout : 1/3 n³

$$
\begin{cases}
r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ik}^2} \\
r_{ji} = \dfrac{1}{r_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} r_{ik} r_{jk} \right) \quad \text{pour } j = i+1 \text{ a } n
\end{cases}
$$

Factorisation QR

- A est inversible
- Trouver Q et R, matrice orthogonale et matrice triangulaire supérieure tel que

$$\mathrm{A} = \mathrm{Q}\,\mathrm{R}$$

- Résoudre le système linéaire équivalent

$$\mathrm{R}\,\boldsymbol{x} = \mathrm{Q}^{\mathrm{T}}\boldsymbol{b}$$

Méthodes directes (fin)

- Matrice symétrique définie positive
  - Conditionnement = Rapport des valeurs propres

$$K(A) = \frac{\lambda_{max}}{\lambda_{min}}$$

- Limite d'utilisation
  - Plus le conditionnement est grand
  - Moins ces méthodes donnent des solutions précises

Deux familles de méthodes

- Méthodes directes

- Méthodes itératives
  - Convergence vers une solution
  - Jacobi
  - Gauss-Seidel
  - Gradient conjugué

Méthodes itératives

- Décomposer A en deux matrices M et N

$$\mathrm{A} = \mathrm{M} - \mathrm{N}$$

- Avancer jusqu'à convergence

$$\mathrm{M}\,\boldsymbol{x}^{(k+1)} = \mathrm{N}\,\boldsymbol{x}^{(k)} + \boldsymbol{b}$$

Méthode de Jacobi

tant que $dist(Ax_{new}, b) > \varepsilon \qquad (i.e.\ 10^{-12})$

$$x_i^{\text{new}} = \frac{b_i - \sum_{j=1, j \neq i}^{n} a_{ij}\, x_j^{\text{old}}}{a_{ii}}$$

fin

Soit D la diagonale de la matrice A, et G le reste :
$A = D+G$

$$D = \begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & a_{ii} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & a_{nn} \end{pmatrix} \qquad G = \begin{pmatrix} 0 & a_{12} & a_{1i} & \cdots & a_{1n} \\ a_{21} & 0 & a_{i-1,i} & \ddots & \vdots \\ a_{i1} & a_{i,i-1} & 0 & a_{i-1,i} & a_{i,n} \\ \vdots & \ddots & a_{i,i+1} & 0 & a_{n-1,n} \\ a_{n1} & \cdots & a_{ni} & a_{n,n-1} & 0 \end{pmatrix}$$

$$Dx^{\text{new}} = b - Gx^{\text{old}} \Leftrightarrow x^{\text{new}} = D^{-1}\left(b - Gx^{\text{old}}\right)$$

méthode de Gauss-Seidel

tant que $dist(Ax_{new}, b) > \varepsilon \qquad (i.e. 10^{-12})$

$$x_i^{\text{new}} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{\text{new}} - \sum_{j=i+1}^{n} a_{ij} x_j^{\text{old}}}{a_{ii}}$$

fin

Soit E la triangulaire inférieure et F la supérieure de la matrice A :

$$A = D + E + F$$

$$E = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \ddots & \vdots \\ a_{i1} & a_{i,i-1} & 0 & 0 & 0 \\ \vdots & \ddots & a_{i,i+1} & 0 & 0 \\ a_{n1} & \cdots & a_{ni} & a_{n,n-1} & 0 \end{pmatrix} \quad F = \begin{pmatrix} 0 & a_{12} & a_{1i} & \cdots & a_{1n} \\ 0 & 0 & a_{i-1,i} & \ddots & \vdots \\ 0 & 0 & 0 & a_{i-1,i} & a_{i,n} \\ \vdots & \ddots & 0 & 0 & a_{n-1,n} \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

$$(D+E)x^{\text{new}} = b - Fx^{\text{old}} \Leftrightarrow x^{\text{new}} = (D+E)^{-1}\left(b - Fx^{\text{old}}\right)$$

Convergence de ces méthodes

- Si A est a diagonale dominante stricte par ligne
  - Les deux convergent.

- Si A est symétriquement définie positive
  - Gauss Seidel converge.

Amélioration itérative

- Supposons une solution numérique au système $Ax=b$
  - La qualité du calculé est estimé par le résidu

$$r = b - Ax_{computed}$$

- Si r est petit (comparé à b), x est précis
- Sinon ?
  - Erreur théorique en x:

$$e = x_{correct} - x_{computed}$$

  - Si on connaît cette erreur, on peut corriger x:

$$x_{correct} = x_{computed} + e$$

  - Système à résoudre pour l'erreur

$$Ax_{computed} = A(x_{correct} - e) = b - r$$
$$Ax_{correct} - Ae = b - r$$
$$Ae = r$$